COM6012 Assignment 1

Question 1. Log Mining

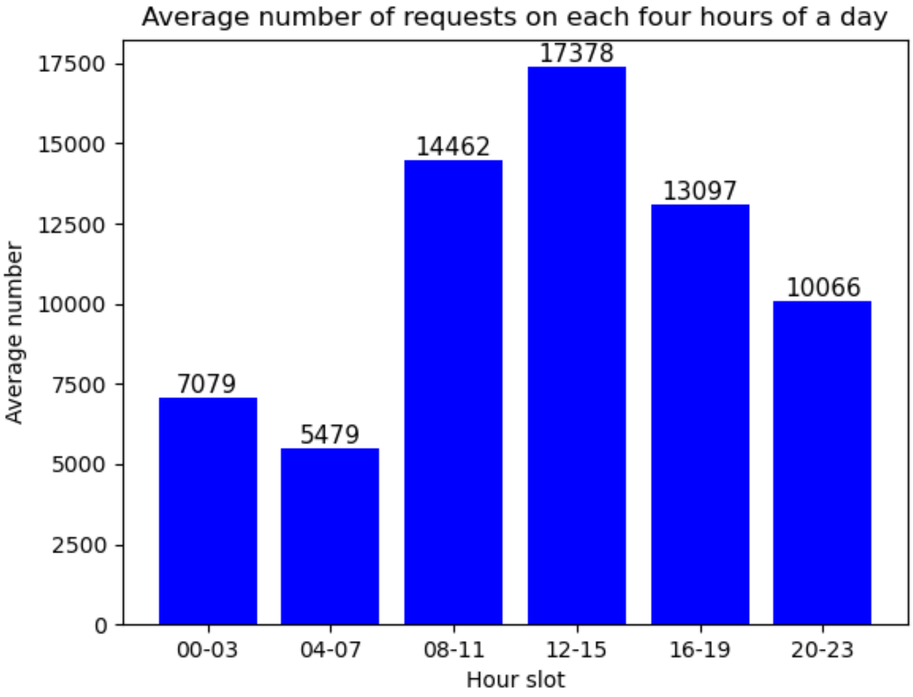
A. The code for this question is in “Q1_code.py” file, and the output is in the “Q1A_output.txt” file generated by the code.

Answer: By loading “NASA_access_log_Jul95.gz” file and using “filter” function, “regexp_extract” function and regular expression to extract data to various time periods. By looking at the original file data, the file recorded 28 days of data in July, so the average value obtained is the total divided by 28 days.

The results are as follows:

Slot	Average request number
00:00:00-03:59:59	7078.96
04:00:00-07:59:59	5479.39
08:00:00-11:59:59	14462.35
12:00:00-15:59:59	17377.78
16:00:00-19:59:59	13096.57
20:00:00-23:59:59	10066.14

B. The code for this question is in “Q1_code.py” file, and the output is in the “Q1B_barchart.png” generated by the code.



Discussion:

1. It can be seen from the chart that in the 24 hours one day, the number of requests is the most from 12 to 15 o’clock, more than 17,000 on average. And the average number of visits at four to seven is the least, only more than 5,000. Therefore, it can be inferred that most users usually access the NASA server at noon.

2. From the observation of the overall trend of the chart, it can be seen that the request number of the NASA server reached the peak of the average value at noon time, and then gradually decreased. In six time periods, the maximum number of requests can reach the minimum three times. Therefore, it's important to keep the server up and running during the day.

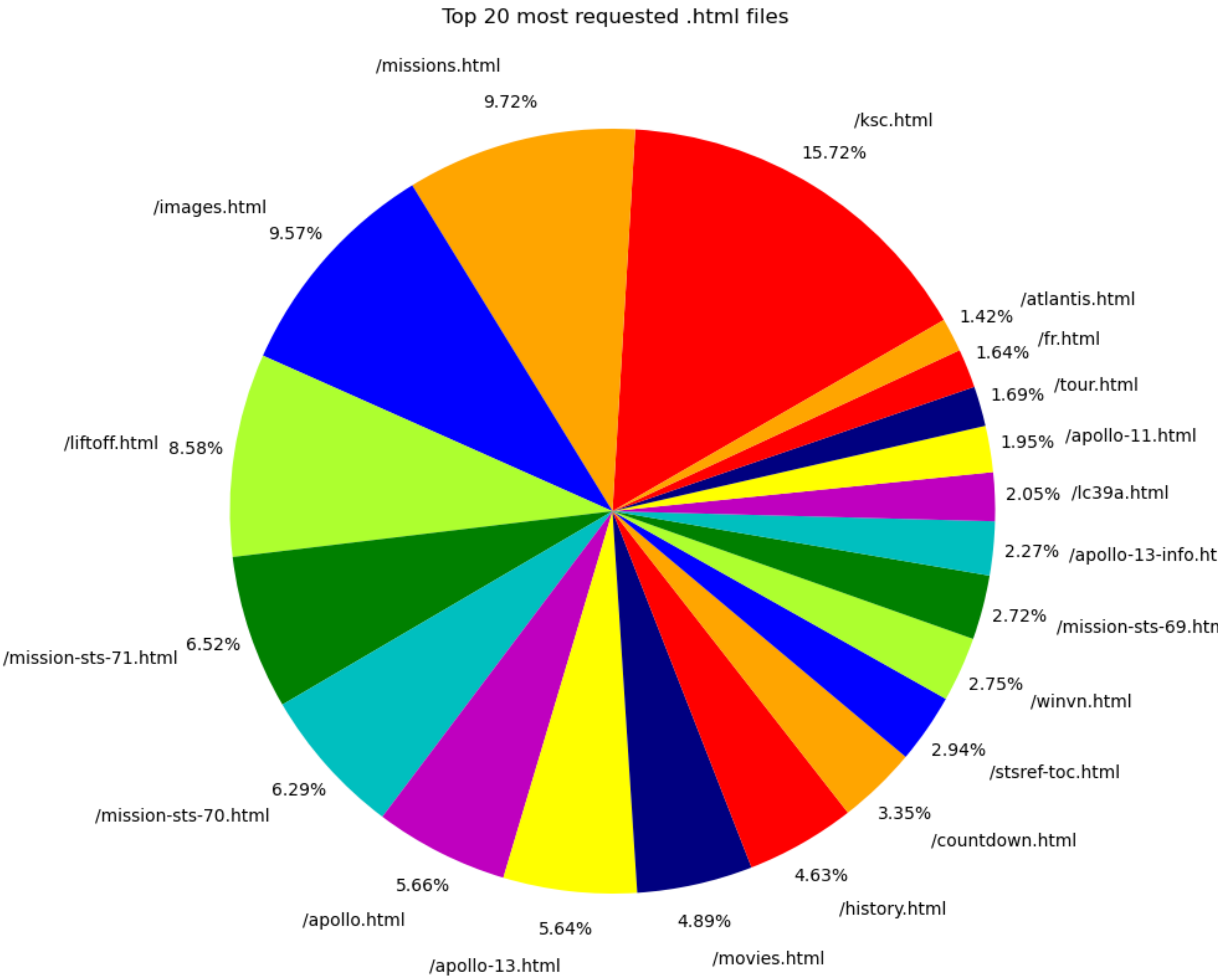
C. The code for this question is in "Q1_code.py" file, and the output is in the "Q1C_output.txt" file generated by the code.

Answer: By loading "NASA_access_log_Jul95.gz" file and using "filter" function, "regexp_extract" function and regular expression to extract html data. Then use "groupby" function to count the html dataframe to get the number of times each file name appears in the log file. Use the "sort" function to sort the counts and output the first 20 file names.

The results are as follows:

1. file name:/ksc.html count:40317
2. file name:/missions.html count:24921
3. file name:/images.html count:24536
4. file name:/liftoff.html count:22012
5. file name:/mission-sts-71.html count:16736
6. file name:/mission-sts-70.html count:16136
7. file name:/apollo.html count:14527
8. file name:/apollo-13.html count:14457
9. file name:/movies.html count:12538
10. file name:/history.html count:11873
11. file name:/countdown.html count:8586
12. file name:/stsref-toc.html count:7538
13. file name:/winvn.html count:7043
14. file name:/mission-sts-69.html count:6987
15. file name:/apollo-13-info.html count:5833
16. file name:/lc39a.html count:5263
17. file name:/apollo-11.html count:5014
18. file name:/tour.html count:4322
19. file name:/fr.html count:4219
20. file name:/atlantis.html count:3640

D. The code for this question is in “Q1_code.py” file, and the output is in the “Q1D_piechart.png” generated by the code.



Discussion:

1. It can be found from the above figure that the most requested file is named “/ksc.html”, which accounts for 15.72% of the top 20 most requested “.html” files. The second is the files named “/missions.html” and “/images.html”, which account for 9.72% and 9.57%, respectively.

2. By observing the top 20 most requested files in the chart, it can be found that the top five most frequently accessed files can account for 50% of the total request number. The last five files have basically the same number of requested.

Question 2. Movie Recommendation

A. The code for this question is in “Q2_all.py” file, and the output is in the “Q2A_output.txt” generated by the code. Visualized chart in “Q2A_barchart.png” file.

Answer: By loading “ratings.csv” file, reading the data in the file and use “randomSplit” function to divide all the data into three subsets of the same size to complete the three-fold cross-validation.

For each split, set three ALS models with parameters as follows:

Model 1: als_1 = ALS(maxIter=10, regParam=0.1, userCol="userId", itemCol="movieId", ratingCol=“rating", coldStartStrategy=“drop")

Model 1 uses default parameters from lab3 notebook.

Model 2: als_2 = ALS(maxIter=20, regParam=0.1, userCol="userId", itemCol="movieId", ratingCol=“rating", coldStartStrategy=“drop")

Model 2 changes the maximum number of iterations to 20.

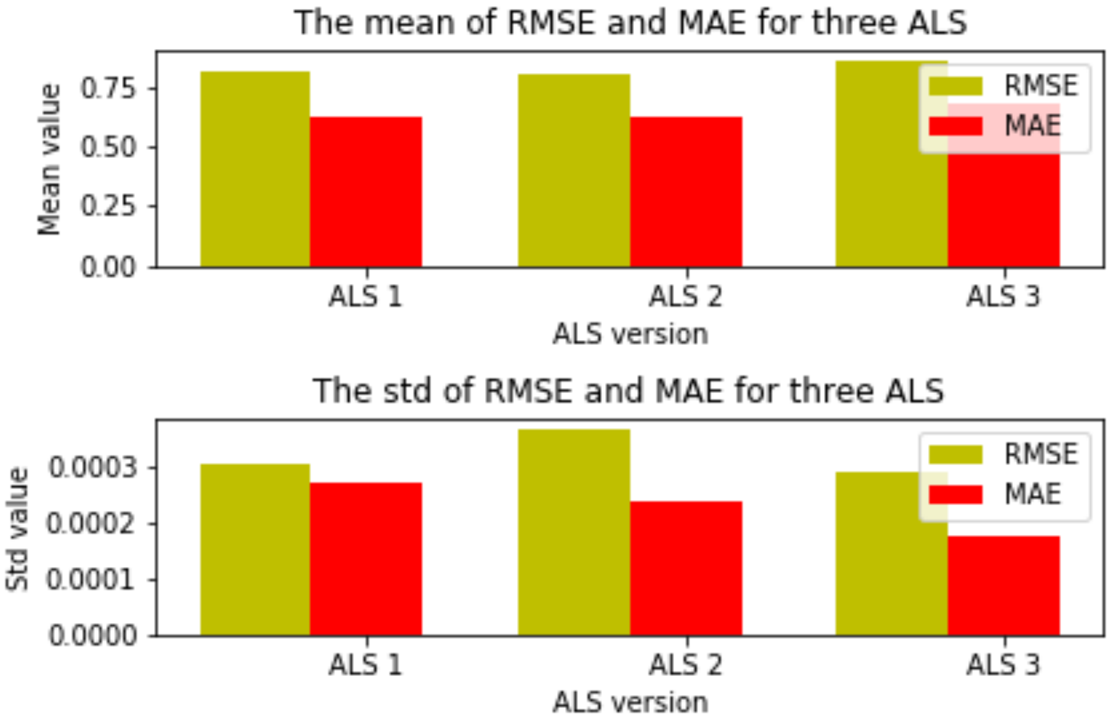
Model 3: als_3 = ALS(maxIter=10, regParam=0.2, userCol="userId", itemCol="movieId", ratingCol=“rating", coldStartStrategy=“drop",nonnegative=True)

Model 3 changes the regularization parameter to 0.2 and sets nonnegative to True.

The results are as follows:

	Split 1		Split 2		Split 3		mean over three splits		std over three splits	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
ALS 1	0.80673	0.62383	0.80661	0.62345	0.80731	0.62411	0.80688	0.62379	0.00030	0.00027
ALS 2	0.80427	0.62092	0.80445	0.62082	0.80512	0.62137	0.80461	0.62104	0.00036	0.00023
ALS 3	0.86014	0.67394	0.86020	0.67386	0.86079	0.67427	0.86038	0.67402	0.00029	0.00017

Visualise the mean and std of RMSE and MAE for each of the three versions of ALS


The mean of RMSE and MAE for three ALS

The std of RMSE and MAE for three ALS

B. Discussion:

1. By observing the data in the table above, it is found that for the three split datasets, the values of RMSE and MAE are very close in the same model parameters, but the results produced by split3 will be larger.

2. By observing the results of three models with different parameters, the RMSE and MAE obtained by model 3 are larger than the other two. It can be seen that changing the regularization parameter from 0.1 to 0.2 will increase the error.

3. For model1 and model2, it can be seen that the maximum number of iterations has a small impact on the error, and the maximum iterations of 20 times can only reduce the error value to a small extent.

C. The code for this question is in “Q2_all.py” file, and the output is in the “Q2C_output.txt” generated by the code.

Answer: After generating three ALS models in 2A, save the models to a list for solving the this question. Firstly, extract the factor of each movie in the model by using “itemFactor” function, and convert the factors to a dense vectors. The second part uses the k-mean model to train the vector and uses the “transform” function to get the cluster value of predictions for each movie id. Thirdly, use the “groupby” function to calculate the data contained in each cluster, and use the “count” function and the “sort” function to sort and extract the top three cluster values. Fourth, the dataframe containing the moiveid and the dataframe containing the prediction value are merged to extract the movieid with the predicted value of the top three clusters, respectively. Fifth, load “genome_score.csv” file to generate a dataframe, merge the moive id extracted in the previous step with the dataframe, extract the tags corresponding to the moivd id in one cluster, and calculate the sum of the value of the relevance of each tag id. The sixth part, load “genome_tags.csv” file, find the tag name corresponding to the top three tag id. Finally, load the “tags.csv” file to filter whether the movieid in each cluster has the extracted tag name and counted.

The results are as follows:

		Cluster num: 0			Cluster num: 2			Cluster num: 3		
Split 1	Tag ID	742	646	468	742	807	646	742	646	270
	Tag name	Original	Mentor	Great ending	Original	Predictable	Mentor	Original	Mentor	Criterion
	Count	127	20	5	0	6	0	7	0	0
		Cluster num: 21			Cluster num: 5			Cluster num: 13		
Split 2	Tag ID	742	646	468	742	807	646	742	270	646
	Tag name	Original	Mentor	Great ending	Original	Predictable	Mentor	Original	Criterion	Mentor
	Count	68	39	111	0	6	1	3	0	0
		Cluster num: 19			Cluster num: 12			Cluster num: 8		
Split 3	Tag ID	742	646	445	742	646	867	742	972	323
	Tag name	Original	Mentor	good	Original	Mentor	Runaway	Original	storytelling	drama
	Count	153	144	13	14	0	2	567	242	765

D. Discussion:

1. By observing each clustering result obtained by each split, it can be seen that the most relevant tag is “tagId=742, tag=‘original”, indicating that most users set the tags for movies as “original”.

2. From the overall observation of the output results, it can be seen that the tag results obtained by each split and the corresponding first cluster are almost the same. This shows that the models obtained by the three split trainings have similar movie labeling factors.

3. By combining the effect of the three splits obtained in 2a on the model error, it can be inferred that when the split 3 is used to train the model, a larger error is obtained. When clustering is performed using the factors of this model, the resulting tag cluster results can also be biased.